ELSEVIER

Theoretical Computer Science 263 (2001) 283–304

Theoretical Computer Science

www.elsevier.com/locate/tcs

A combinatorial approach to Golomb forests ☆

Mordecai J. Golin

Department of Computer Science, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, People's Republic of China

Accepted April 2000

Abstract

Optimal binary prefix-free codes for infinite sources with geometrically distributed frequencies, e.g., $\mathscr{P} = \{p^i(1-p)\}_{i=0}^{\infty}$, $0 < p < 1$, were first (implicitly) suggested by Golomb over 30 years ago in the context of run-length encodings. Ten years later Gallager and Van Voorhis exhibited such optimal codes for all values of p. These codes were derived by using the Huffman encoding algorithm to build optimal codes for *finite* sources and then showing that the finite codes converge in a very specific sense to the infinite one. In this note, we present a new combinatorial approach to solve the same problem, one that does not use the Huffman algorithm, but instead treats a coding tree as an infinite sequence of integers and derives properties of the sequence. One consequence of this new approach is a complete characterization of *all* of the optimal codes; in particular, it shows that for all $p, 0 < p < 1$, except for an easily describable countable set, there is a unique optimal code, but for each p in this countable set there are an *uncountable* number of optimal codes. Another consequence is a derivation of infinite codes for geometric sources when the encoding alphabet is no longer restricted to be the binary one. A final consequence is the extension of the results to optimal forests instead of being restricted to optimal trees.

1. Introduction

Consider an information source $\mathscr{P}$ with probability distribution

$$\mathscr{P} = \{p_i\}_{i=0}^{n-1}, \quad p_0 \geqslant p_1 \geqslant p_2 \geqslant \cdots \geqslant p_{n-1}$$

on a set of n letters. The *Huffman Encoding problem* is to associate a prefix-free set of n binary words $\{w_i\}_{i=0}^{n} \subset \{0,1\}^*$ with $\mathscr{P}$ such that the expected word length $\sum_{i=0}^{n} p_i \,\text{length}(w_i)$ is minimized, where $\text{length}(w_i)$ is the number of bits in w_i, e.g., $\text{length}\,(0110) = 4$. A *prefix-free* set is one in which $\forall i \neq j$, w_i is not a prefix of w_j.

☆ This research was partially supported by Hong Kong RGC/CRG grant HKUST652/95E.
E-mail address: golin@cs.ust.hk (M.J. Golin).

0304-3975/01/$ - see front matter
PII: S0304-3975(00)00250-4

It is well known (see e.g., [14]), that finding such a code is equivalent to finding a tree with n leaves such that when l_i is the length of the ith highest leaf in the tree then the expected external path length $\sum_{i=0}^{n} p_i l_i$ (achieved by placing the ith letter (p_i) at the ith highest leaf) is minimized. Such a tree may easily be found using the well-known *Huffman Encoding Algorithm* [6].

Suppose now that the situation is modified slightly to permit *infinite* sources, i.e.,

$$\mathscr{P} = \{p_i\}_{i=0}^{\infty}, \quad p_0 \geqslant p_1 \geqslant p_2 \geqslant \cdots .$$

In this case the problem of finding a prefix-free code, or equivalently, an infinite tree labeled with the p_i, with minimum weighted external path length, is not nearly as well understood. It has been proven [11] that optimal trees (codes) exist if and only if the entropy $-\sum_i p_i \log p_i$, of $\mathscr{P}$ is bounded [1] but there still does not exist any algorithm for finding optimal codes that works for all such $\mathscr{P}$ with bounded entropy.

Special cases are better understood, though. The best known and earliest such case is that of the infinite binary codes (e.g., using only 0-s and 1-s) for the infinite geometric source. This is the source that fixes some p, $0 < p < 1$, and then defines $\mathscr{P}_p = \{(1-p)p^i\}_{i=0}^{\infty}$. Such a source arises, for example, in the description of run-length encoding as was noted by Golomb [4]. Suppose we have a string of **A**s and **B**s in which each character occurs independently of every other one; **B**s occurring with probability p and **A**s with probability $1-p$. Now, for $i = 0, 1, 2, \ldots$ set $X_i = \underbrace{BB\ldots BB}_{i \text{ times}}A$. *Every* infinite string can be written uniquely as the concatenation of different X_is with the probability of X_i occurring being $(1-p)p^i$. We thus, have a situation in which strings are composed of words from an infinite source with given distribution $\mathscr{P}_p$. Other problems that can be recast as finding a min-cost infinite tree with distribution $\mathscr{P}_p$ arise in operations research [5] and group testing [8, 15].

This special case of $\mathscr{P} = \mathscr{P}_p$ was studied by Gallager and Van Voorhis [3] who exhibited an optimal tree [2] for every p. Their technique was to 'guess' a countable sequence of *finite* sources $\mathscr{P}_p^0$, $\mathscr{P}_p^1$, $\mathscr{P}_p^2$, $\mathscr{P}_p^3, \ldots$, that were better and better approximations to $\mathscr{P}_p$, use the Huffman algorithm to derive the optimal trees for these finite sources and then show that these codes "converge" to an infinite tree that is optimal for the infinite source. Their result can be stated as:

Theorem 1 (Gallager and Van Voorhis [3]). *Given p, let m be the unique integer that satisfies*

$$p^m + p^{m+1} \leqslant 1 < p^m + p^{m-1}.$$

[1] We refer the reader to [9] for a discussion of how to generalize the concept of optimality so that "optimal" infinite Huffman codes exist even when the entropy is unbounded.

[2] The codes associated with these trees are sometimes known as *Golomb codes*. Hence the identification of these trees as Golomb trees and the natural extension to forests as Golomb forests.

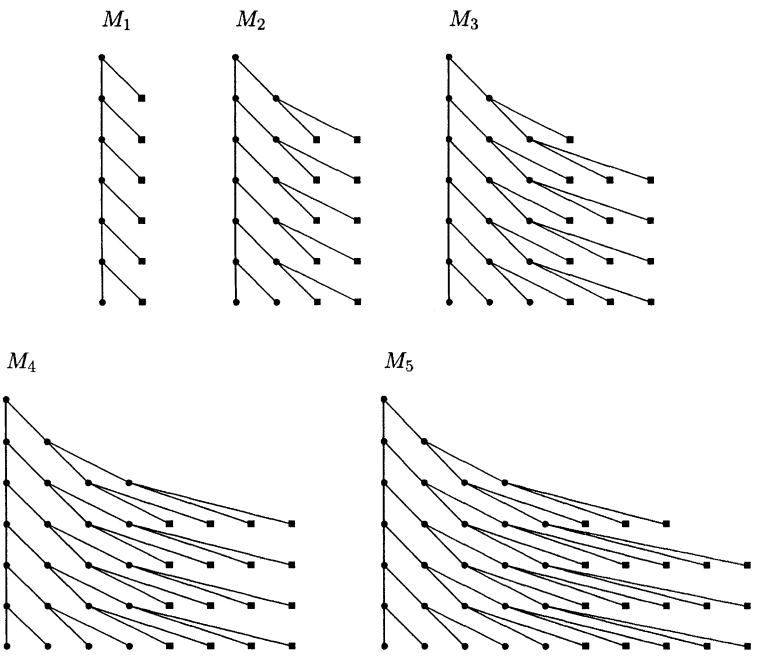

Fig. 1. The tops of the infinite "optimal" trees M_1, M_2, M_3, M_4 and M_5.

Let a tree T be described as a sequence I_i, $i = 0, 1, 2, 3, \ldots$, where I_i is the number of internal nodes on level i. Then the tree described by

$$I_0,\ I_1,\ I_2,\ I_3, \ldots, = 1,\ 2,\ 4, \ldots, 2^{\lfloor \log_2 m \rfloor}, m,\ m,\ m, \ldots \tag{1}$$

is optimal for $\mathscr{P}_p$.

If we use M_m to denote the mth such tree, then Fig. 1 contains the tops of M_1, M_2, M_3, M_4 and M_5. In this figure, as in all the others in this paper, a filled-in circle represents an internal node while a square represents a leaf.

Gallager and Van Voorhis' basic technique was later expanded upon and extended by Abrahams [1] and Kato et al. [10] who showed that it could be applied to other families of infinite sources. Abrahams [1] also showed how to extend the theorem to cover the ternary case in which every parent can have three children rather than two. There are also some very new results that describe how to extend the results to two-sided geometric distributions [12, 13]. A comprehensive survey of the latest results may be found in [2].

All of these papers share the same basic approach, in that they construct an optimal code/tree for the *infinite* source by using the Huffman encoding algorithm to construct optimal trees/codes for a sequence of special *finite* sources and then showing that these finite trees/codes "converge" in a nice fashion to an infinite one which must be an optimal tree/code for the infinite source.

In this paper, we return to the original Gallager and Van Voorhis problem of the geometric source and provide a new derivation of optimal binary trees/codes for geometric sources by deriving the structure of all optimal forests. This new derivation does not use the Huffman algorithm to build finite trees or forests. It instead treats an optimal infinite forest as an infinite sequence of integers (that represent the number of leaves/internal-nodes on each level of the forest) and proves combinatorial properties of such sequences, e.g., their elements are bounded, after some point the sequence must cycle, etc. These properties will permit a complete description of the structure of optimal trees (as opposed to the old proofs which exhibited only one optimal tree but said nothing about the existence or nonexistence of any others). It will also permit generalizing the results in [3] from binary trees to d-ary forests,[3] where the generalization is both from binary to d-ary and from trees to forests.

The rest of this paper is structured as follows. Section 2 introduces some basic definitions, translating problems on trees and forests into problems on equivalent infinite sequences. Section 3 presents the main theorem which completely characterizes the structure of optimal trees for geometric sources. Section 4 introduces some basic combinatorial operations on the tree sequences and shows that many of them preserve optimality. Section 5 pulls everything together and proves the main theorem. Section 6 discusses an interesting related problem left open.

2. Definitions

We start by noting that as far as our problem is concerned the actual topological structure of a tree is not important; for calculating costs the only important quantities in a tree are the numbers of *leaves* at each of its levels. Any two trees that have identical numbers of leaves at all levels will have the same cost. For our purposes it will also be convenient to know the number of *internal nodes* at each level and to be able to represent forests (collections of trees).

Also, for reasons to be discussed later in this section, this paper will only be concerned with *full* forests, forests in which each internal node has a full complement of d children. Therefore, for all $l \geqslant 0$, $E_{l+1} + I_{l+1} = dI_l$. We thus define:

Definition 1. Let the arity, $d \geqslant 2$ be fixed. A *forest* $F = \{(I_l, E_l)\}_{l=0}^{\infty}$ is an infinite sequence of pairs of nonnegative integers satisfying

$$\forall l \geqslant 0, \quad E_{l+1} + I_{l+1} = dI_l.$$

The I_l are called *internal* nodes while the E_l are called *external* or *leaf* nodes. Given a forest F we define

$$\forall l \geqslant 0, \quad I_l(F) = I_l \quad \text{and} \quad E_l(F) = E_l.$$

[3] We should point out that even though [3] restricts itself to binary trees its proof technique could be modified to also derive optimal binary forests.

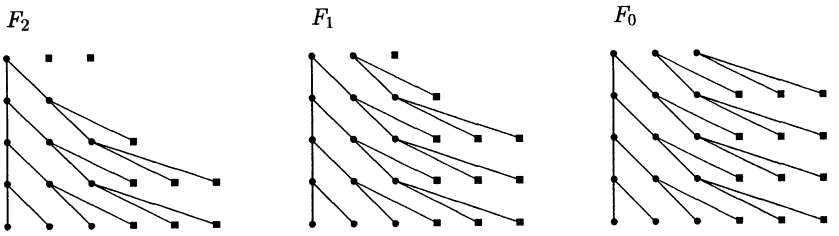

Fig. 2. The tops of three infinite binary forests that all have the same number of nodes on their upper levels.

For pedagogical reasons, we also define $A_l(F) = \sum_{j \leqslant l} E_j(F)$ where $A_l(F)$ is the number of *leaves* on or above level l. When F is understood we will simply write A_l. Finally, a *tree* T is a forest with $(I_0(T), E_0(T)) = (1, 0)$.

Since $E_{l+1} + I_{l+1} = dI_l$ a forest is fully determined by knowing just the number of leaves on its top level and the number of internal nodes on all levels. Thus, we will usually write a forest as a sequence:

$$F = \langle E_0;\ I_0,\ I_1,\ I_2,\ I_3, \ldots \rangle.$$

For example the forests F_2, F_1 and F_0 in Fig. 2 have initial sequences

$$F_2 = \langle 2; 1, 2, 3, 3, 3, 3, \ldots \rangle,$$

$$F_1 = \langle 1; 2, 3, 3, 3, 3, 3, \ldots \rangle \quad \text{and} \quad F_0 = \langle 0; 3, 3, 3, 3, 3, 3, \ldots \rangle.$$

We now define $\mathrm{Cost}(\cdot, \cdot)$ in a way that corresponds to the natural cost of a tree given a source.

Definition 2. Let F be a forest and $d_i(F)$, $i = 0, 1, 2, \ldots$ be the depth of the ith leaf in F, breaking ties arbitrarily. Thus,

$$d_i(F) = \min\{l:\ i < A_l(F)\}.$$

Let $\mathscr{P} = \{p_i\}_{i=0}^{\infty}$ be a nonincreasing sequence of nonnegative reals. The *cost of F labeled by P* is the external path length of forest F when its leaves, sorted by increasing depth, are labeled with the p_i, i.e.,

$$\mathrm{Cost}(F, \mathscr{P}) = \sum_{0 \leqslant i} p_i d_i(F) = \sum_{0 \leqslant i} p_i \min\{l : i < A_l(F)\}.$$

Note: if $\sum_l E_l(F)$ is bounded then set $\mathrm{Cost}(F, \mathscr{P}) = \infty$ (this describes the degenerate case in which all but a finite number of nodes in the forest are internal. Such a forest obviously cannot be labeled with an infinite number of leaves).

As an example, the forest F_1 in the diagram has

$$\mathrm{Cost}(F_1, \mathscr{P}) = 0\,p_0 + 1\,p_1 + 2\,p_2 + 2\,p_3 + 2\,p_4 + 3\,p_5 + 3\,p_6 + \cdots.$$

Definition 3. A forest F is *optimal* for $\mathscr{P}$ if it has minimal cost among all forests with the same number of nodes on their top levels, i.e., it is optimal if

$$\forall \text{Forests } F' \text{ such that } I_0(F') + E_0(F') = I_0(F) + E_0(F),$$

$$\text{Cost}(F, \mathscr{P}) \leqslant \text{Cost}(F', \mathscr{P}).$$

Similarly, A tree T is optimal for $\mathscr{P}$ if

$$\forall \text{Trees } T', \quad \text{Cost}(T, \mathscr{P}) \leqslant \text{Cost}(T', \mathscr{P}).$$

As previously mentioned it is known [11] that optimal trees (and, by slightly modifying their proof, optimal forests as well) actually do exist for every $\mathscr{P}$ with finite entropy $H(P) = -\sum_i p_i \ln p_i$.

We note that an optimal forest must be full because, if it was not, adding the "missing" node(s) as a leaf (leaves) would result in a forest with lesser cost. For the rest of this paper we will, therefore, always assume that forests are full and, in particular, obey the equations, $E_{l+1} + I_{l+1} = dI_l$.

We also note that, given a fixed forest F, it is impossible to permute the elements of $\mathscr{P}$ to create a new sequence $\mathscr{P}'$ containing the same elements in a different order such that $\text{Cost}(F, \mathscr{P})' \leqslant \text{Cost}(F, \mathscr{P})$. This follows directly from the fact that the elements in $\mathscr{P}$ are sorted in nonincreasing order with $p_0 \geqslant p_1 \geqslant p_2 \geqslant \cdots$. Thus, the optimality of a forest is not only over all forests with the same number of nodes on their top level but also over all permutations of $\mathscr{P}$.

It is quite easy to see that scaling $\mathscr{P}$ by a constant does not change optimality: for every $\alpha > 0$, F is optimal for P if and only if F is optimal for

$$\alpha P = \alpha p_0, \alpha p_1, \alpha p_2, \ldots .$$

We, therefore, will not restrict ourselves to $\mathscr{P}$ for which $\sum_i p = 1$ because after finding an optimal tree for any general $\mathscr{P}$, we can always go back and scale it so that its elements sum to 1.

In what follows, we will always assume that $0 < p < 1$ is fixed and $\mathscr{P}_p = \{p^i\}_{i=0}^{\infty}$, is the geometric sequence it generates. We will thus abbreviate $\text{Cost}(F, \mathscr{P})$ to $\text{Cost}(F)$.

Finally, note that for geometric series the cost has a particularly simple form:

Lemma 1. *If p is fixed then*

$$\text{Cost}(F, \mathscr{P}) = \sum_{0 \leqslant i} p_i \cdot \min\{l : i < A_l(F)\} = \sum_{0 \leqslant l} \sum_{A_l \leqslant j} p^j = \frac{1}{1-p} \sum_{0 \leqslant l} p^{A_l(F)}.$$

As an example note that in Fig. 2, if all levels in F_1 below the second level have three leaves, then

$$\text{Cost}(F_1, \mathscr{P}) = \frac{1}{1-p}\left(p + p^2 \sum_{0 \leqslant i} p^{3i}\right) = \frac{p(1 + p - p^3)}{(1-p)(1-p^3)}.$$

3. The main result

As previously mentioned, the main result of this note is a combinatorial derivation of the structure of optimal codes for geometric sources. An advantage of this new derivation (as opposed to the existing ones) is that it permits the identification of *all* of the optimal Huffman trees. In particular, it shows that for all but a countable number of ps there is a *unique* optimal tree, while for each of that countable number of ps there are an *uncountable* but, still easily describable, set of optimal trees. More specifically:

Definition 4. Fix the arity $d \geqslant 2$. Set $\alpha_0^{(d)} = 0$ and for $m > 0$ define $\alpha_m^{(d)}$ to be the unique positive real root of

$$1 - p^{m(d-1)} \left(\sum_{i=0}^{d-1} p^i \right) = 0.$$

Note that $\forall m > 0,\ 0 < \alpha_m^{(d)} < 1$ and $\alpha_m^{(d)} \uparrow 1$. The main result of this paper is:

Theorem 2 (Structure Theorem). *Let the number of nodes on the top level of a forest be fixed to be some $I \geqslant 1$.*

(1) *If $\alpha_{m-1}^{(d)} < p < \alpha_m^{(d)}$ then there exists a unique optimal forest F for p and it has the form*

$$F = \begin{cases} \langle 0;\ I,\ dI,\ d^2I, \ldots, d^{\lfloor \log_d m/I \rfloor} I,\ m,\ m, \ldots \rangle & \text{if } I \leqslant m, \\ \langle I - m;\ m,\ m,\ m, \ldots, \ldots \rangle & \text{if } I > m. \end{cases} \tag{2}$$

(2) *If $p = \alpha_m^{(d)}$ then let $\mathscr{S} = \{m, m+1\}^{\mathscr{N}}$ be the set of all infinite tuples that can be written using integers m and $m+1$. For $S \in \mathscr{S}$ we write $S = (S_0, S_1, S_2, \ldots)$ and define F_S to be the forest*

$$F_S = \begin{cases} \langle 0;\ I,\ dI,\ d^2I, \ldots, d^{\lfloor \log_d m/I \rfloor} I,\ S_0,\ S_1,\ S_2,\ S_3, \ldots \rangle & \text{if } I \leqslant m, \\ \langle I - S_0;\ S_0,\ S_1,\ S_2,\ S_3, \ldots \rangle & \text{if } I > m. \end{cases}$$

Then, the set of optimal forests for p is exactly equal to $\{F_S\colon S \in \mathscr{S}\}$.

This theorem says that, given p and the number of nodes on the top level of the forest, then if $p \neq \alpha_m^{(d)}$ for some m there exists exactly one unique forest, while if $p = \alpha_m^{(d)}$ then there are an uncountable number of forests.

The rest of this paper is dedicated in proving this theorem using combinatorial tools. Before continuing, we note that when $d = 2$ the trees ($I = 1$) defined in Eq. (2) are what are called M-codes in [1]. Also note that the $\alpha_m^{(2)}$ are the roots of $1 - p^m - p^{m+1}$ so

$$p \leqslant \alpha_m^{(2)} \Leftrightarrow 1 - p^m - p^{m+1} \geqslant 0 \Leftrightarrow 1 \geqslant p^m + p^{m+1}.$$

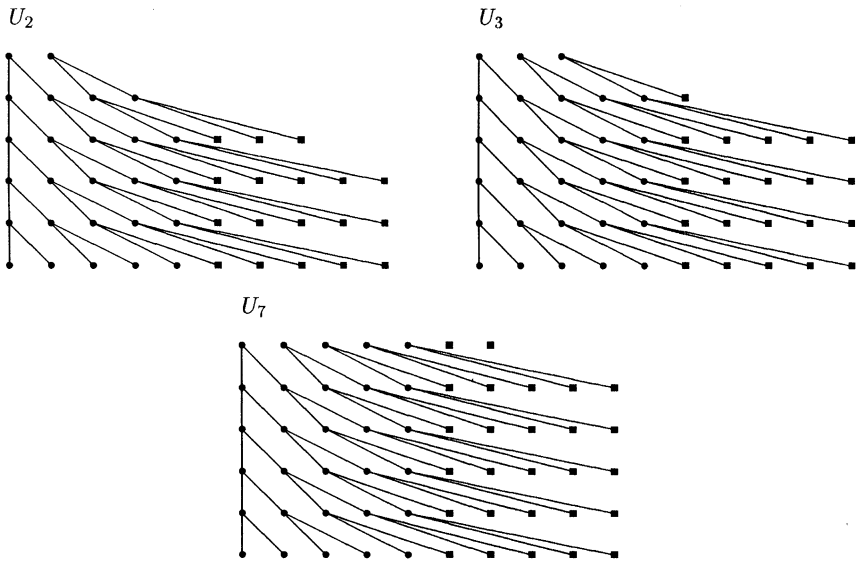

Fig. 3. If $\alpha_4^{(2)} < p < \alpha_5^{(2)}$ then the forests U_2, U_3 and U_7 are the *only* optimal forests with, respectively, 2, 3 and 7 nodes on the top level. Note that the unshown levels of the forests all have exactly five internal nodes and 5 leaves.

This in turn implies that

$$\alpha_{m-1}^{(2)} < p \leqslant \alpha_m^{(2)} \Leftrightarrow p^m + p^{m+1} \leqslant 1 < p^m + p^{m-1}.$$

Thus Theorem 2 implies Theorem 1.

For other examples of the applications of the theorem we refer the reader to Fig. 3, which illustrates some unique optimal binary nontree forests; Fig. 4 which illustrates two of the possible optimal binary trees for $p = \alpha_4^{(2)}$; and Fig. 5 which illustrates three optimal ternary ($d = 3$) trees. We note that in Fig. 4 the tree S' has the form

$$S' = \langle 0;\ 1,\ 2,\ 4,\ 3,\ 4,\ 3,\ 3,\ 4,\ 3,\ 3,\ 3,\ 4, \ldots \rangle,$$

where each successive 'run' of 3s is one longer than its predecessor.

4. Combinatorial operations on forests

In this section we define some basic combinatorial operations that can be performed on forests and state facts about them.

In what follows we assume that p is fixed and that

$$F = \langle E_0;\ I_0,\ I_1,\ I_2,\ I_3, \ldots \rangle \quad \text{and} \quad F' = \langle E_0';\ I_0',\ I_1',\ I_2',\ I_3', \ldots \rangle$$

are forests. Recall that, by definition, $E_{l+1} + I_{l+1} = dI_l$ and $E_{l+1}' + I_{l+1}' = dI_l'$.

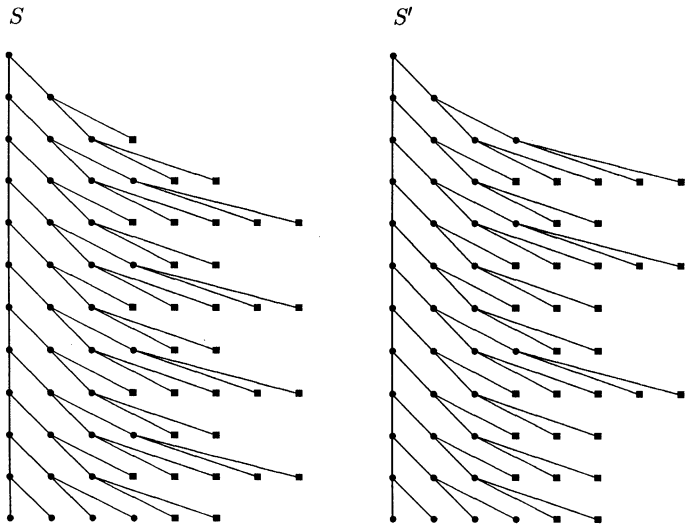

Fig. 4. The trees shown here are both tops of infinite trees that are optimal for $p=\alpha_4^{(2)}$.

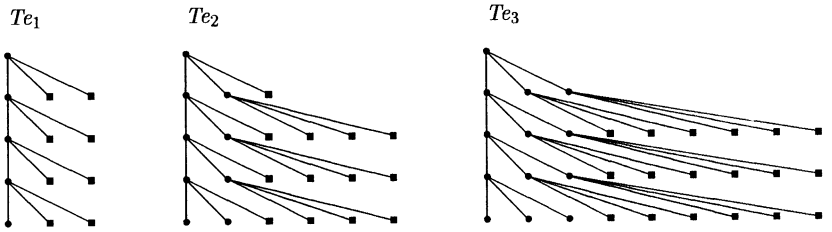

Fig. 5. The tops of the "optimal" ternary trees Te_1, Te_2, Te_3. Te_1 is optimal if $0<p\leqslant\alpha_1^{(3)}$; Te_2 is optimal if $\alpha_1^{(3)}\leqslant p\leqslant\alpha_2^{(3)}$; Te_3 is optimal if $\alpha_2^{(3)}\leqslant p\leqslant\alpha_3^{(3)}$.

Definition 5. Let $i,j\geqslant 0$ and $m>0$ be integers.

$$F^{\langle i\rangle}=\langle i;\ I_0,\ I_1,\ I_2,\ I_3,\ldots\rangle,$$

$$\text{Trunc}(F,j)=\langle E_j;\ I_j,\ I_{j+1},\ I_{j+2},\ldots\rangle,$$

$$\text{Repeat}(F,j)=\langle E_0;\ I_0,\ I_1,\ I_2,\ldots,I_{j-1},\ I_j,\ I_j,\ I_{j+1},\ I_{j+2},\ldots\rangle,$$

$$\text{Cycle}(F,j)=\langle E_0;\ I_0,\ I_1,\ I_2,\ldots,I_{j-1},\ I_j,\ I_j,\ I_j,\ I_j,\ldots\rangle,$$

$$C_m=\langle 0;\ m,\ m,\ m,\ m,\ldots\rangle.$$

Suppose further that for some j, $I'_0+E'_0=I_j+E_j$. Then Replace (F,F',j) is defined so that

$$\forall i<j,\ I_i(\text{Replace}(F,F',j))=I_i,\quad E_i(\text{Replace}(F,F',j))=E_i,$$

$$\forall i\geqslant j,\quad I_i(\text{Replace}(F,F',j))=I'_{i-j},\ E_i(\text{Replace}(F,F',j))=E'_{i-j},$$

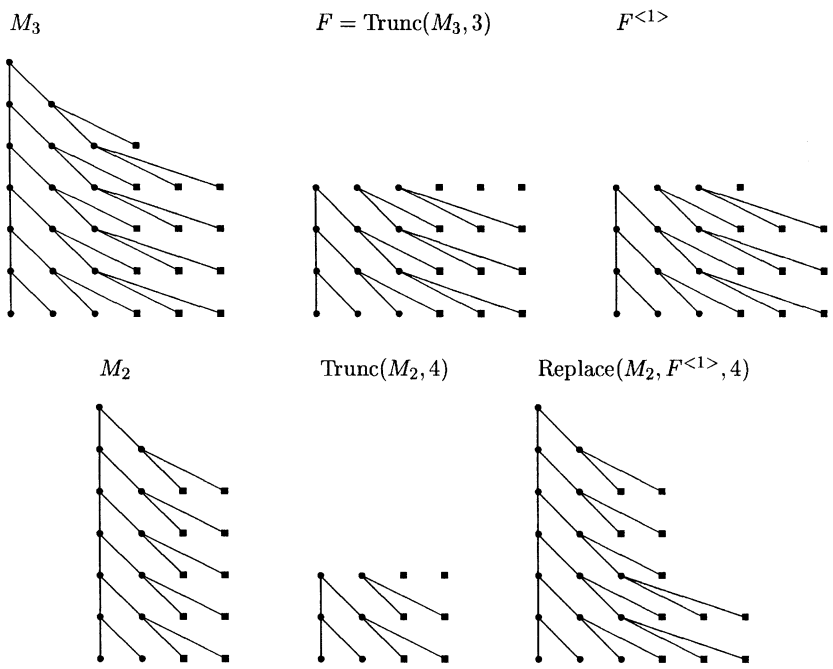

Fig. 6. The trees M_2 and M_3 are as described previously. The other forests are examples of the operations introduced in Definition 5. For illustration purposes forests defined using the Trunc( ) operator are left on the same level that they appeared on their original trees.

i.e.,

$$\text{Replace}(F,\ F',\ j) = \langle E_0;\ I_0,\ I_1,\ I_2, \ldots, I_{j-2},\ I_{j-1},\ I'_0,\ I'_1,\ I'_2,\ I'_3, \ldots \rangle.$$

$F^{\langle i \rangle}$ is Forest F with i instead of E_0 leaves on its top level. $\text{Trunc}(F, j)$ is F starting at level j. $\text{Repeat}(F, j)$ is F with I_j repeated *once*. $\text{Cycle}(F, j)$ is the top j levels of F and then I_j repeated forever. C_m is just the forest with m internal nodes on every level. Note that, by definition, $C_{I_j} = \text{Trunc}(\text{Cycle}(F, j), j)^{\langle 0 \rangle}$. $\text{Replace}(F, F', j)$ starts with F, keeps its top j levels, and replaces everything on level j and below with forest F'. Examples of these operations can be found in Figs. 6–8. Also C_1 is just tree M_1 from Fig. 1 and C_4 can be found in Fig. 7.

Many of these operations preserve optimality of forests.

Lemma 2. *If F and F' are optimal for p then*
(1) $\forall j \geqslant 0$, $\text{Trunc}(F, j)$ *is optimal.*
(2) *If* $E'_0 + I'_0 = E_j + I_j$ *then* $\text{Replace}(F, F', j)$ *is optimal.*
(3) $\forall i \leqslant E_0$, $F^{\langle i \rangle}$ *is optimal.*
(4) *If* $I'_0 \leqslant I_0 \leqslant I'_0 + E'_0$ *then* $\forall i \leqslant I'_0 + E'_0 - I_0$, $F^{\langle i \rangle}$ *is optimal.*

Before proving this lemma we describe a typical application. Referring to Fig. 6 suppose that both M_2 and M_3 are optimal. From Lemma 2(1) we have that both

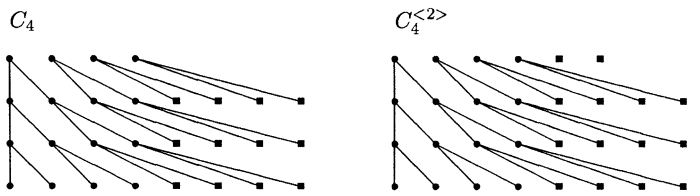

Fig. 7. The trees C_4 and $C_4^{\langle 2\rangle}$. Given that $C_4^{\langle 2\rangle}$ and $F = \mathrm{Trunc}(M_3, 3)$ are optimal Lemma 2(4) immediately implies that $C_4^{\langle 2\rangle}$ is also optimal. Note that Lemma 2(3) *cannot* be used to prove this fact.

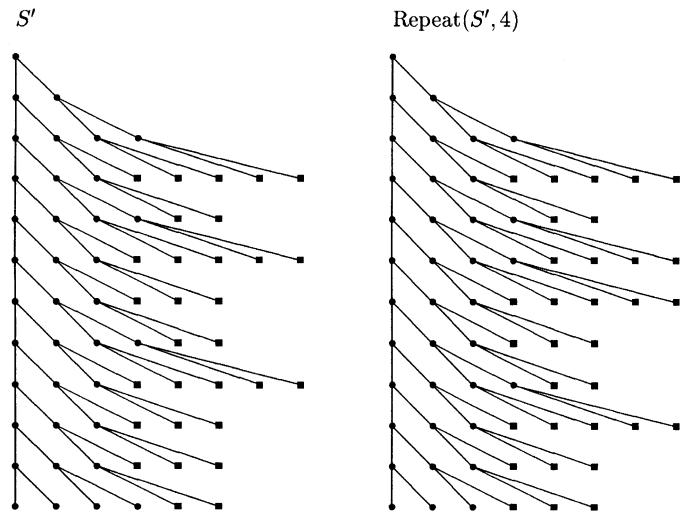

Fig. 8. The tree S' seen previously and Repeat$(S', 4)$. Note that the four internal nodes on level 4 of S' are repeated on level 5 of Repeat$(S', 4)$ after which the remainder of the levels of S' are continued.

$F = \mathrm{Trunc}(M_3,\ 3)$ and $\mathrm{Trunc}(M_2,\ 4)$ are optimal. Since F is optimal, from Lemma 2(3) we have that $F^{\langle 1\rangle}$ is also optimal. Since both $\mathrm{Trunc}(M_2,\ 4)$ and $F^{\langle 1\rangle}$ have exactly the same number (four) of nodes on their top level we have from Lemma 2(2) that $\mathrm{Replace}(M_2,\ F^{\langle 1\rangle},\ 4)$ is also optimal.

Proof. F can be thought of as having two parts; a top part consisting of levels $0, 1, \ldots, j-1$, and a bottom part consisting of $\mathrm{Trunc}(F, j)$. $\mathrm{Replace}(F, F', j)$ is constructed by keeping the top part of F but replacing the bottom part with F': thus,

$$\mathrm{Cost}(F) - \mathrm{Cost}(\mathrm{Replace}(F, F', j)) = p^{A_{j-1}}(\mathrm{Cost}(\mathrm{Trunc}(F, j)) - \mathrm{Cost}(F')). \quad (3)$$

Now suppose that Lemma 2(1) is not true and that, for some j, $\mathrm{Trunc}(F, j)$ is not optimal. Let F' be any optimal forest with $E'_0 + I'_0 = E_j + I_j$. Then $\mathrm{Cost}(F') <$

Cost(Trunc(F,j)) so Eq. (3) implies

$$\text{Cost}(\text{Replace}(F,\ F',\ j)) < \text{Cost}(F)$$

contradicting the optimality of F. This proves Lemma 2(1).

To prove Lemma 2(2) note that if F' is optimal with $E'_0 + I'_0 = E_j + I_j$ then the optimality of Trunc$(F,\ j)$ from Lemma 2(1) implies that $\text{Cost}(F') = \text{Cost}(\text{Trunc}(F,\ j))$ so Eq. (3) implies

$$\text{Cost}(\text{Replace}(F,\ F',\ j)) = \text{Cost}(F)$$

and the optimality of Replace$(F,\ F',\ j)$.

We will now prove that if $E_0 > 0$ then $F^{\langle E_0-1\rangle}$ is optimal. The proof of Lemma 2(3) will follow by iterating this fact to show that $F^{\langle E_0-2\rangle}, F^{\langle E_0-3\rangle}, \ldots, F^{\langle 0\rangle}$ are all optimal as well.

So assume, by contradiction, that $F^{\langle E_0-1\rangle}$ is not optimal and let F' be some optimal forest with $I'_0+E'_0 = I_0+E_0-1$ nodes on its top level. Then $\text{Cost}(F') < \text{Cost}(F^{\langle E_0-1\rangle}) = p^{-1}\,\text{Cost}(F)$. Now add one new leaf to the top level of F' to find

$$\text{Cost}(F'^{\langle E'+1\rangle}) = p\,\text{Cost}(F') < \text{Cost}(F).$$

But, since $F'^{\langle E'_0+1\rangle}$ has the same number of nodes on its top level as F does this contradicts the optimality of F. Thus, $F^{\langle E_0-1\rangle}$ is optimal and Lemma 2(3) follows.

To prove Lemma 2(4) first note that from Lemma 2(3) both $F^{\langle 0\rangle}$ and $F'^{\langle I_0-I'_0\rangle}$ are optimal forests with I_0 nodes on their top levels. Then,

$$\begin{aligned}\text{Cost}(F^{\langle I'_0+E'_0-I_0\rangle}) &= p^{I'_0+E'_0-I_0}\text{Cost}(F^{\langle 0\rangle})\\ &= p^{I'_0+E'_0-I_0}\text{Cost}(F'^{\langle I_0-I'_0\rangle})\\ &= \text{Cost}(F').\end{aligned}$$

Since F' is optimal so is $F^{\langle I'_0+E'_0-I_0\rangle}$. Another application of Lemma 2(3) shows that for every $i \leqslant I'_0 + E'_0 - I_0$, $F^{\langle i\rangle}$ is also an optimal forest, proving Lemma 2(4). □

The last lemma provides us with tools for manipulating optimal trees. For example,

Lemma 3. *Let* $F = \langle E_0; I_0, I_1, I_2, I_3, \ldots\rangle$ *be an optimal forest for* p *such that, for some* j, $I_{j+1} \leqslant I_j$. *Then both* Repeat(F,j) *and* Repeat$(F,j+1)$ *are also optimal forests.*

Proof. Set $F_j = \text{Trunc}(F,j)$ and $F_{j+1} = \text{Trunc}(F,j+1)$. From Lemma 2(1) we have that both F_j and F_{j+1} are optimal forests. Note that

$$I_0(F_j) = I_j, \qquad E_0(F_j) = E_j,$$
$$I_0(F_{j+1}) = I_{j+1}, \quad E_0(F_{j+1}) = E_{j+1}.$$

In particular, since

$$I_{j+1} \leqslant I_j < dI_j = I_{j+1} + E_{j+1}$$

we have

$$I_0(F_{j+1}) \leqslant I_0(F_j) < I_0(f_{j+1}) + E_0(F_{j+1}).$$

Thus, by Lemma 2(4) $F_j^{\langle (d-1)I_j \rangle}$ is also an optimal forest. Finally, applying Lemma 2(2) shows that

$$\text{Repeat}(F, j) = \text{Replace}(F, F_j^{\langle (d-1)I_j \rangle}, j+1)$$

is also optimal. To prove Repeat$(F, j+1)$ simply note that

$$E_0(F_{j+1}) = dI_j - I_{j+1} \geqslant (d-1)I_{j+1}$$

so, by Lemma 2(3), $F_{j+1}^{\langle (d-1)I_{j+1} \rangle}$ is an optimal forest and thus

$$\text{Repeat}(F, j+1) = \text{Replace}(F, F_{j+1}^{\langle (d-1)I_{j+1} \rangle}, j+2)$$

is also optimal. □

As a concrete application, refer back to Fig. 8. In that figure the tree S' has $I_4 = 4 < 3 = I_3$. Thus, if S' is optimal then Repeat$(S', 4)$ is also optimal.

Actually, we can prove a much stronger result, specifically, that if $I_j \geqslant I_{j+1}$ then Cycle(F, j) and Cycle$(F, j+1)$ are both optimal.

First note the following lemma which says that if a sequence of optimal forests 'converges' level-by-level to some forest F then F is also optimal.

Lemma 4 (Convergence Lemma). *Let F be a forest and F_j, $j = 0, 1, 2, \ldots$, be some sequence of optimal forests such that F_j is identical to F on its first j levels, i.e., $\forall j$, $E_0(F_j) = E_0$ and $\forall l < j$, $I_l(F_j) = I_l$. Then F is also an optimal forest.*

Proof. Let C be the cost of an optimal forest with $E_0 + I_0$ nodes on its top level. Then $\forall j$, $\text{Cost}(F_j) = C$. The conditions of the lemma imply that $\forall i < A_j(F)$, $d_i(F_j) = d_i(F)$. Thus, by the definition of cost, we also have

$$C_j = \sum_{i < A_j} p_i \, d_i(F_j) = \sum_{i < A_j} p_i \, d_i(F) < C,$$

where C_j is the cost contributed by the leaves on the first j levels of F_j. But, again by the definition of cost,

$$\text{Cost}(F) = \lim_j C_j \leqslant C.$$

Since $\text{Cost}(F) \geqslant C$ (definition of optimal cost C) this implies $\text{Cost}(F) = C$ and the optimality of F. □

Corollary 5. *Let F be an optimal forest for p and j such that $I_{j+1} \leqslant I_j$. Then the forests C_{I_j} and $C_{I_{j+1}}$ are both optimal.*

Proof. Iteratively define F_i as follows: $F_0 = F$ and $\forall i > 0$, $F_i = \text{Repeat}(F_{i-1}, j)$, e.g.,

$$F_0 = \langle E_0; I_0, I_1, I_2, \ldots, I_{j-1}, I_j, I_{j+1}, I_{j+2}, \ldots \rangle,$$
$$F_1 = \langle E_0; I_0, I_1, I_2, \ldots, I_{j-1}, I_j, I_j, I_{j+1}, I_{j+2}, \ldots \rangle,$$
$$F_2 = \langle E_0; I_0, I_1, I_2, \ldots, I_{j-1}, I_j, I_j, I_j, I_{j+1}, I_{j+2}, \ldots \rangle,$$
$$F_3 = \langle E_0; I_0, I_1, I_2, \ldots, I_{j-1}, I_j, I_j, I_j, I_j, I_{j+1}, I_{j+2}, \ldots \rangle.$$

By Lemma 3, all of the F_i are optimal. The proof that $\text{Cycle}(F, j)$ is an optimal that follows from Lemma 4. Since $C_{I_j} = \text{Trunc}(F, j)^{\langle 0 \rangle}$ the optimality of C_{I_j} then follows from Lemma 2(1) and (3).

The proof that $\text{Cycle}(F, j+1)$ and thus, $C_{I_{j+1}} = \text{Trunc}(F, j+1)^{\langle 0 \rangle}$ are optimal is similar. □

5. Proof of the main theorem

Corollary 5 says that if F is optimal and, for some j, $I_{j+1} \leqslant I_j$ then $\text{Cycle}(F, j)$ and $\text{Cycle}(F, j+1)$ are both optimal trees. A priori, there is no reason to expect that such a j exists, though; perhaps the I_j are monotonically increasing. The next lemma implies that such a j always exists.

Lemma 6 (Optimal forests have bounded width). *Let p be fixed and $B = \min\{k : p^k < 1 - p\}$. Then, if F is any optimal forest, $\forall l, I_l(F) \leqslant B$.*

Proof. Suppose by contradiction that F is optimal and $I_l(F) > B$ for some l. Without loss of generality we may assume that $l = 0$ and $E_l = 0$. Otherwise, replace F by the optimal forest $\text{Trunc}(F, l)^{\langle 0 \rangle}$.

Now note that since all leaves of F are below level 0, $1/(1-p) < \text{Cost}(F)$. But then

$$\text{Cost}(C_1^{\langle I_0(F)-1 \rangle}) = \frac{p^{I_0(F)-1}}{(1-p)^2} \leqslant \frac{p^B}{(1-p)^2} \leqslant \frac{1}{1-p} < \text{Cost}(F)$$

contradicting the optimality of F. □

Recall the definition of the *Cyclic Forest*

$$C_m = \langle 0; m, m, m, m, m, \ldots \rangle. \tag{4}$$

Note that if F is optimal for some P there must *always exist* some j such that $I_{j+1} \leqslant I_j$. Otherwise the I_i are a monotonically increasing sequence, contradicting Lemma 6. For this j both $\text{Cycle}(F, j)$ and $\text{Cycle}(F, j+1)$ are optimal and thus $C_{I_j} = \text{Trunc}(F, j)^{\langle 0 \rangle}$ and $C_{I_{j+1}} = \text{Trunc}(F, j+1)^{\langle 0 \rangle}$ are also both optimal.

Since the C_m have such special forms we can actually calculate for which p they are optimal.

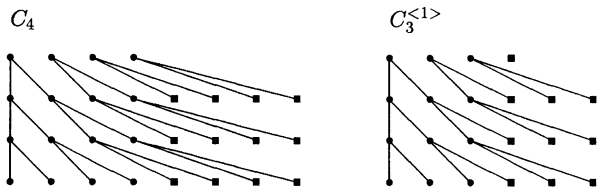

Fig. 9. The trees C_4 and $C_3^{\langle 1\rangle}$, both of which have four nodes on their top levels.

Lemma 7. *Let C_m be as defined above. Then*

$$\mathrm{Cost}(C_m^{\langle 1\rangle}) \leqslant \mathrm{Cost}(C_{m+1}) \tag{5}$$

if and only if

$$p \leqslant \alpha_m^{(d)} \tag{6}$$

with equality in (5) *if and only if there is equality in* (6).

Proof. Refer to Fig. 9. From Lemma 1 we find that

$$\begin{aligned}\mathrm{Cost}(C_{m+1}) &= \frac{1}{1-p}(1 + p^{(m+1)(d-1)} + p^{2(m+1)(d-1)} + \cdots)\\ &= \frac{1}{(1-p)}\frac{1}{(1-p^{(m+1)(d-1)})}\end{aligned}$$

and

$$\mathrm{Cost}(C_m^{\langle 1\rangle}) = \frac{1}{1-p}(p + p^{m(d-1)+1} + p^{2m(d-1)+1} + \cdots) = \frac{1}{(1-p)}\frac{p}{(1-p^{m(d-1)})}.$$

Thus,

$$\begin{aligned}\mathrm{Cost}(C_{m+1}) - \mathrm{Cost}(C_m^{\langle 1\rangle}) &= \frac{1}{1-p}\left(\frac{1}{1-p^{(m+1)(d-1)}} - \frac{p}{1-p^{m(d-1)}}\right)\\ &= \frac{1 - p^{m(d-1)} - p + p^{(m+1)(d-1)+1}}{(1-p)(1-p^{(m+1)(d-1)})(1-p^{m(d-1)})}\\ &= \frac{(1-p) - p^{m(d-1)}(1-p^d)}{(1-p)(1-p^{(m+1)(d-1)})(1-p^{m(d-1)})}\\ &= \frac{1 - p^{m(d-1)}\sum_{0\leqslant i<d} p^i}{(1-p^{(m+1)(d-1)})(1-p^{m(d-1)})}.\end{aligned}$$

The proof of the lemma follows from Definition 4 in which $\alpha_m^{(d)}$ is defined to be the unique positive real root of $1 - p^{m(d-1)}(\sum_{i=0}^{d-1} p^i)$. □

Corollary 8. *C_m is optimal if and only if $\alpha_{m-1}^{(d)} \leqslant p \leqslant \alpha_m^{(d)}$.*

Proof. From the discussion preceding Lemma 5 we know that for any fixed p there must exist at least one m such that C_m is optimal for p (e.g., $m=I_j$ where j is such that $I_{j+1}\leqslant I_j$). We now use Lemma 5 to discover what the possible values of m are.

First suppose that $p<\alpha_{k-1}^{(d)}$. Then $\mathrm{Cost}(C_{k-1}^{\langle 1\rangle})<\mathrm{Cost}(C_k)$ so C_k is not optimal.

Now suppose that $\alpha_k^{(d)}<p$. Then $\mathrm{Cost}(C_k^{\langle 1\rangle})>\mathrm{Cost}(C_{k+1})$ so $C_k^{\langle 1\rangle}$ is not optimal. But if C_k was optimal then from Lemma 2 $C_k^{\langle 1\rangle}=\mathrm{Trunc}(C_k,1)^{\langle 1\rangle}$ is also optimal.

We can condense the above paragraphs into two statements:

$$\text{If } p<\alpha_{k-1}^{(d)} \quad \text{then } C_k \text{ is not optimal for } p. \tag{7}$$

$$\text{If } \alpha_k^{(d)}<p \quad \text{then } C_k \text{ is not optimal for } p. \tag{8}$$

We now prove the lemma. First suppose that p is such that $\alpha_{t-1}^{(d)}<p<\alpha_t^{(d)}$ for some t. From (7) we have that $t-1<m$ while from (8) we have that $m<t+1$. In other words, C_m is optimal for $m=t$ and no other m.

Now suppose that $p=\alpha_t^{(d)}$ for some t. Then the same reasoning shows that $t-1<m<t+2$ or that $m\in\{t,t+1\}$ so at least one of C_t and C_{t+1} must be optimal but if $m\notin\{t,t+1\}$ then C_m is not optimal. Suppose first that C_{t+1} is optimal. From Lemma 7, $\mathrm{Cost}(C_t^{\langle 1\rangle})=\mathrm{Cost}(C_{t+1})$ so $C_t^{\langle 1\rangle}$ is optimal and thus, from Lemma 2, C_t itself is also optimal.

Now suppose that C_t is optimal. Then from Lemma 2, $C_t^{\langle 1\rangle}=\mathrm{Trunc}(C_t,1)^{\langle 1\rangle}$ is also optimal. From Lemma 7, $\mathrm{Cost}(C_{t+1})=\mathrm{Cost}(C_t^{\langle 1\rangle})$ and thus C_{t+1} is also optimal.

We have just seen that if $p=\alpha_t^{(d)}$ then C_t is optimal if and only if C_{t+1} is optimal. Thus, they are both optimal, completing the proof of the corollary. □

We need one more corollary before proceeding. It says that if $p=\alpha_m^{(d)}$ then there are an uncountable number of optimal forests:

Corollary 9. *Let $\mathscr{S}=\{m,m+1\}^{\mathcal{N}}$ be the set of all infinite tuples that can be written using integers m and m+1. For $S\in\mathscr{S}$ we write $S=(S_0,S_1,S_2,\ldots)$. Then, $\forall S\in\mathscr{S}$ define*

$$U_S=\langle 0;S_0,S_1,S_2,S_3,\ldots\rangle.$$

Then $\forall S\in\mathscr{S}$, U_S is optimal for $p=\alpha_m^{(d)}$. Furthermore, $\forall i\leqslant(d-1)(m+1)$, $\forall S\in\mathscr{S}$, $U_S^{\langle i\rangle}$ is also optimal.

Proof. In the previous corollary we have already seen that

$$C_m=\langle 0;m,m,m,m,\ldots\rangle \quad\text{and}\quad C_{m+1}=\langle 0;m+1,m+1,m+1,m+1,\ldots\rangle$$

are both optimal for $p=\alpha_m^{(d)}$. Straightforward application of Lemma 2 shows that

$$V_m=C_m^{\langle(d-1)m+d\rangle} \quad\text{and}\quad V_{m+1}=C_{m+1}^{\langle(d-1)m-1\rangle}$$

are also both optimal. Now recursively define

$$F_0=C_{S_0} \quad\text{and}\quad \forall i>0,\ F_i=\mathrm{Replace}(F_{i-1},V_{S_i},i).$$

That is

$$F_0 = \langle 0; S_0, S_0, S_0, S_0, S_0, S_0, S_0, \ldots \rangle,$$

$$F_1 = \langle 0; S_0, S_1, S_1, S_1, S_1, S_1, S_1, \ldots \rangle,$$

$$F_2 = \langle 0; S_0, S_1, S_2, S_2, S_2, S_2, S_2, \ldots \rangle,$$

$$F_3 = \langle 0; S_0, S_1, S_2, S_3, S_3, S_3, S_3, \ldots \rangle,$$

$$F_4 = \langle 0; S_0, S_1, S_2, S_3, S_4, S_4, S_4, \ldots \rangle.$$

Note that, again from Lemma 2 we find that all of the F_i are optimal forests. Setting $F = U_S$ and applying Lemma 4 proves that $U_S = F$ is also optimal.

To prove that $U_S^{\langle i \rangle}$ is optimal for $i \leqslant (d-1)(m+1)$ note that the proof above also shows that

$$U_{(m+1,\ S)} = \langle 0; m+1, S_0, S_1, S_2, S_3, \ldots \rangle$$

is also optimal. Then Lemma 2 shows that

$$U_S^{\langle i \rangle} = \mathrm{Trunc}(U_{(m+1,\ S)}, 1)^{\langle i \rangle}$$

is also optimal. □

We can now *almost* prove the main theorem. Specifically, we can prove:

Lemma 10. *Let $F = \langle E_0; I_0, I_1, I_2, I_3, \ldots \rangle$ be an optimal forest. Let k be the smallest value for which $I_k \leqslant I_{k+1}$, i.e,*

$$I_0 < I_1 < I_2 < \cdots < I_k \geqslant I_{k+1}.$$

Then,

- *if $\alpha_{m-1}^{(d)} < p < \alpha_m^{(d)}$ then $\forall j \geqslant k,\ I_j = m$,*
- *if $p = \alpha_m^{(d)}$ then $\forall j \geqslant k,\ I_j \in \{m, m+1\}$.*

Proof. Let j be any value such that $I_{j+1} \leqslant I_j$. From Lemma 6 we know that such a j exists. From Corollary 5 we have that forests C_{I_j} and $C_{I_{j+1}}$ are both optimal. But from Corollary 8 we see that only this can happen

- if $\alpha_{m-1}^{(d)} < p < \alpha_m^{(d)}$ and $I_j = I_{j+1} = m$,
- or if $p = \alpha_m^{(d)}$ and $I_j, I_{j+1} \in \{m, m+1\}$.

The proof follows. □

We now know that $I_0 < I_1 < I_2 < \cdots < I_k \geqslant I_{k+1}$ and how I_j behaves for $j \geqslant k$. We require one more technical lemma that will be used to derive how I_j grows when $j < k$.

Lemma 11. *Let $\alpha_{m-1}^{(d)} < p \leqslant \alpha_m^{(d)}$. If F is a forest with $E_0 = 1$ and $I_0 < m$ then F is not optimal for p.*

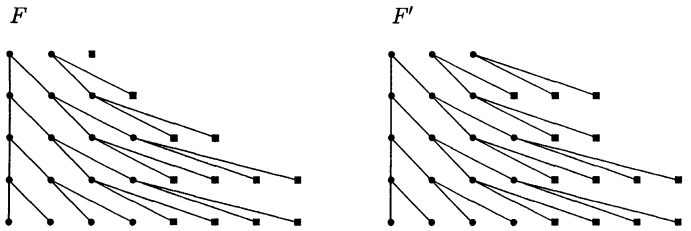

Fig. 10. Example trees F and F' for the proof of Lemma 11. Note that F' is created by taking the leaf on the top level of F and making it internal. In this example we assume that $m=4$.

Proof. We will assume that such an F is optimal and show a contradiction.

We will also assume that F satisfies

$$I_0<I_1<I_2<\cdots<I_{k-1}<m=I_k=I_{k+1}=I_{k+2}=\cdots \tag{9}$$

for some k. If $\alpha_{m-1}^{(d)}<p<\alpha_m^{(d)}$ then Lemma 10 says F necessarily must satisfy this condition. If $p=\alpha_m^{(d)}$ then it is possible that F might not satisfy the condition. Lemma 10 says that F still must be of the form

$$I_0<I_1<I_2<\cdots<I_k\geqslant I_{k+1}$$

with $\forall j\geqslant k, I_j\in\{m,m+1\}$ and that the forest with the same E_0 value and

$$I_0<I_1<I_2<\cdots<I_{k-1}<m=I_k=I_{k+1}=I_{k+2}=\cdots$$

is also optimal. We can then take this new optimal forest as our F.

So now assume that we have an optimal $F=\langle 1; I_0, I_1, I_2, I_3,\ldots\rangle$ satisfying (9) with $E_0=1$ and $I_0<m$. Let $F'=\langle 0; I_0+1, I_1, I_2, I_3,\ldots\rangle$ be the forest resulting from transforming the highest leaf of F into an internal node all of whose d children are leaves (see for, example, Fig. 10). We will now show that $\mathrm{Cost}(F')<\mathrm{Cost}(F)$ contradicting the optimality of F and completing the proof.

Recall that $A_i(F)$ is the number of leaves in F on or above level i and that $\mathrm{Cost}(F)=1/(1-p)\sum_{0\leqslant l}p^{A_l(F)}$. Note that $A_0(F)=1$, and $A_0(F')=0$. Also,

$$\forall i>0,\quad A_i(F')=A_i(F)+d-1.$$

Thus,

$$\mathrm{Cost}(F)=\frac{1}{1-p}\left(p+\sum_{1\leqslant l}p^{A_l(F)}\right)$$

$$\begin{aligned}\mathrm{Cost}(F')&=\frac{1}{1-p}\left(1+\sum_{1\leqslant l}p^{A_l(F')}\right)\\&=\frac{1}{1-p}\left(1+p^{d-1}\sum_{1\leqslant l}p^{A_l(F)}\right)\end{aligned}$$

so

$$\begin{aligned}\mathrm{Cost}(F') - \mathrm{Cost}(F) &= \frac{1}{1-p}\left(1 - p - [1 - p^{d-1}]\sum_{1\leqslant l} p^{A_l(F)}\right)\\ &= 1 - \left(\sum_{0\leqslant i<d-1} p_i\right)\sum_{1\leqslant l} p^{A_l(F)}.\end{aligned}$$

Now note that $I_0 < I_1 \leqslant m$ so $I_0 \leqslant m-1$. Therefore,

$$E_1 = dI_0 - I_1 < (d-1)I_0 \leqslant (d-1)(m-1).$$

Since $E_0 = 1$ this implies that

$$A_1(F) = E_0 + E_1 \leqslant 1 + (d-1)(m-1) - 1 = (d-1)(m-1).$$

Also note that since $\forall i \geqslant 0$ we have $I_i \leqslant I_{i+1}$ and $I_i \leqslant m$,

$$E_{i+1} = dI_i - I_{i+1} \leqslant (d-1)I_i \leqslant (d-1)m.$$

But $A_{i+1}(F) = A_i(F) + E_i$ so iterating shows that

$$\forall i \geqslant 1, \quad A_i \leqslant (d-1)(m-1) + (i-1)(d-1)m.$$

Summing over i gives

$$\sum_{1\leqslant l} p^{A_l(F)} \geqslant \sum_{1\leqslant l} p^{(d-1)(m-1)+(l-1)(d-1)m} = \frac{p^{(d-1)(m-1)}}{1-p^{(d-1)m}}.$$

Plugging back into our cost equations yields

$$\begin{aligned}\mathrm{Cost}(F') - \mathrm{Cost}(F) &= 1 - \left(\sum_{0\leqslant i<d-1} p^i\right)\sum_{1\leqslant l} p^{A_l(F)}\\ &\leqslant 1 - \left(\sum_{0\leqslant i<d-1} p^i\right)\frac{p^{(d-1)(m-1)}}{1-p^{(d-1)m}}\\ &= \frac{1 - p^{(d-1)m} - (\sum_{0\leqslant i<d-1} p^i)p^{(d-1)(m-1)}}{1-p^{(d-1)m}}\\ &= \frac{1 - (\sum_{0\leqslant i<d} p^i)p^{(d-1)(m-1)}}{1-p^{(d-1)m}}.\end{aligned}$$

But, since $\alpha_{m-1}^{(d)} < p$ we know that

$$1 - p^{(d-1)(m-1)}\sum_{0\leqslant i<d} p^i < 0$$

and $\mathrm{Cost}(F') < \mathrm{Cost}(F)$, contradicting the optimality of F. □

Corollary 12. *Let* $\alpha_{m-1}^{(d)} < p \leqslant \alpha_m^{(d)}$. *If* F *is an optimal forest for* p *then*

(1) *If* $I_0 < m$ *then* $E_0 = 0$.

(2) $\forall i \geqslant 0,\ I_{i+1} \geqslant \min\{dI_i, m\}$.

Proof. To prove (1) note that if $I_0 < m$ but $E_0 > 0$ then, by Lemma 2 $F^{\langle 1\rangle}$ is optimal contradicting Lemma 11.

To prove (2) note that if, for some i, $I_{i+1} < \min\{dI_i, m\}$ then $E_{i+1} = dI_i - I_{i+1} > 0$. Thus Lemma 2 says that $\mathrm{Trunc}(F,i)^{\langle 1\rangle}$ is optimal. But since $I_{i+1} < m$ Lemma 11 says that $\mathrm{Trunc}(F,i)^{\langle 1\rangle}$ is not optimal, leading to a contradiction. □

We can now prove Theorem 2. In what follows, we assume that $F = \langle E_0; I_0, I_1, I_2, I_3, \ldots\rangle$ is an optimal tree for p and examine its possible structures.

We examine the theorem's two cases of $\alpha_{m-1}^{(d)} < p < \alpha_m^{(d)}$ and $p = \alpha_m^d$ separately:

$\alpha_{m-1}^{(d)} < p < \alpha_m^{(d)}$. Before starting, note that Lemma 10 implies that $\forall i,\ I_i \leqslant m$. In particular $I_0 \leqslant m$. On the other hand, Corollary 12 can be read as saying $I_0 \geqslant \min\{I, m\}$. We thus find that $I_0 = \min\{I, m\}$.

If $I \leqslant m$ set $l = \lfloor \log_d m/I \rfloor$. From the argument above we have that $I_0 = I$. Successive applications of Corollary 12 show that

$$I_0, I_1, I_2, I_3, \ldots, I_l = I, dI, d^2I, \ldots, d^lI.$$

Since $dI_l = d^{l+1}I > m$ Corollary 12 implies that $I_{l+1} \geqslant m$. Lemma 10 then says that $\forall i > l,\ I_i = m$. Combining everything yields

$$F = \langle 0; I, dI, d^2I, \ldots, d^{\lfloor \log_d m/I\rfloor}I, m, m, \ldots\rangle.$$

If $I > m$ then $I_0 = m$ and thus, by Lemma 10 $\forall i > 0,\ I_i = m$. In other words

$$F = \langle I - m; m, m, m, \ldots, \ldots\rangle.$$

$p = \alpha_m^{(d)}$. The beginning of the analysis is very similar to that performed in the previous case. Note here that Lemma 10 implies that $\forall i,\ I_i \leqslant m + 1$. In particular $I_0 \leqslant m+1$. On the other hand, Corollary 12 can be read as saying $I_0 \geqslant \min\{I, m\}$. We thus find that if $I \leqslant m$ then $I_0 = I$ while if $I > m$ then $I_0 \in \{m, m+1\}$.

Now suppose that $I \leqslant m$. We have seen that $I_0 = I$. As before, successive applications of Corollary 12 show that

$$I_0, I_1, I_2, I_3, \ldots, I_l = I, dI, d^2I, \ldots, d^lI.$$

Using Lemma 10 we then find that $\forall i > l,\ I_i \in \{m, m+1\}$. Combining everything yields

$$F = \langle 0; I, dI, d^2I, \ldots, d^{\lfloor \log_d m/I\rfloor}I, S_0, S_1, S_2, S_3, \ldots\rangle$$

for some $S = (S_0, S_1, S_2, \ldots) \in \mathscr{S}$.

Now let $S' = (S'_0, S'_1, S'_2, \ldots) \in \mathscr{S}$ be *any* $S' \in \mathscr{S}$. Since $d^lI \leqslant m$ and $S'_0 \in \{m, m+1\}$ we have that $d^{l+1}I - S_0 \leqslant (d-1)m$. Thus, from Lemma 9 we find that $U_{S'}^{\langle d^{l+1}I - S'_0\rangle}$ is

optimal. Since this forest has d^{l+1} nodes on its top level and F_S has $dI_l = d^{l+1}$ nodes on its $(l+1)$st level Lemma 2 says that

$$F_{S'} = \text{Replace}(F_S, U_{S'}^{\langle d^{l+1} I - S_0' \rangle}, l+1)$$

is also optimal. In other words, we have just shown that a tree F is optimal if and only if $F = F_S$ for some $S \in \mathscr{S}$.

Suppose then that $I > m$. We have just seen that $I_0 > m$, $I_0 \in \{m, m+1\}$. Lemma 10 then tells us that $\forall i > 0$, $I_i \in m, m+1$. Thus,

$$F = F_S = \langle I - S_0; S_0, S_1, S_2, S_3, \ldots \rangle$$

for some $S = (S_0, S_1, S_2, \ldots) \in \mathscr{S}$. Note that in the notation of Corollary 9 $F = F_S = U_S^{\langle I - S_0 \rangle}$.

We now show that

$$F_{S'} = \langle I - S_0'; S_0', S_1', S_2', S_3', \ldots \rangle$$

is optimal for all $S' = (S_0', S_1', S_2', \ldots) \in \mathscr{S}$.

First note that if $S_0 = S_0'$ then, by Corollary 9 both U_S and $U_{S'}$ are optimal. Since they both have the same number $(S_0 = S_0')$ of nodes on their top level this implies $\text{Cost}(U_S) = \text{Cost}(U_{S'})$ so

$$\text{Cost}(U_S^{\langle I - S_0 \rangle}) = p^{I - S_0} \text{Cost}(U_S) = p^{I - S_0'} \text{Cost}(U_{S'}) = \text{Cost}(U_{S'}^{\langle I - S_0 \rangle}).$$

But, since $F_{S'} = U_{S'}^{\langle I - S_0' \rangle}$ this says $\text{Cost}(F_S) = \text{Cost}(F_{S'})$ so $F_{S'}$ is also optimal.

Now suppose that $S_0' \neq S_0$. Without loss of generality we will assume that $S_0' = m$ and $S_0 = m+1$ (the other case is symmetric). From Corollary 9 both U_S and $U_{S'}^{\langle 1 \rangle}$ are optimal. Since they both have $m+1$ nodes on their top level this implies $\text{Cost}(U_S) = \text{Cost}(U_{S'}^{\langle 1 \rangle})$ so

$$\text{Cost}(U_S^{\langle I - S_0 \rangle}) = p^{I - (m+1)} \text{Cost}(U_S) = p^{I-m} \text{Cost}(U_{S'}^{\langle 1 \rangle}) = \text{Cost}(U_{S'}^{\langle I - S_0 \rangle}).$$

But, since $F_{S'} = U_{S'}^{\langle I - S_0' \rangle}$ this says $\text{Cost}(F_S) = \text{Cost}(F_{S'})$ so $F_{S'}$ is also optimal. In other words we have just shown that a tree F is optimal if and only if $F = F_S$ for some $S \in \mathscr{S}$ completing the proof of Theorem 2.

6. Conclusion

In this paper we derived combinatorial properties of optimal (minimum-external path length) forests for distributions $\mathscr{P}_p = \{p^i(1-p)\}_{i=0}^{\infty}$. These combinatorial properties permitted us to exactly derive the form of these optimal forests.

One very interesting open question would be the construction of such trees for other distributions. At the moment the only distributions for which optimal trees are known seem to be the geometric ones, some of its variations [12, 13] and the (tails

of) Poisson distributions [7]. Others have not been addressed. It would, for example, be quite interesting to know the optimal tree for the Zeta-distributions

$$P_\alpha = \left\{ \frac{1}{i^\alpha} \right\}_{i=1}^{\infty},$$

where $\alpha > 1$. We note that one complication that arises in the optimal trees for these derivations is that their *width*, i.e., the maximum number of nodes that can appear on any level is unbounded. This is in sharp contrast to the situation in the geometric case in which Lemma 6 bounds (as a function of p) the number of nodes that can appear on any level.

Acknowledgements

The author would like to thank Julia Abrahams and Akiko Kato for providing pointers and references to this problem. He would also like to thank Yong Xue Rong and the anonymous referees for careful comments on an earlier version of this paper.

References

[1] J. Abrahams, Huffman-type codes for infinite source distributions, J. Franklin Inst. 331B (3) (1994) 265–271.
[2] J. Abrahams, Code and parse trees for lossless source encoding, Sequences 1997 (1997).
[3] R.G. Gallager, D.C. Van Voorhis, Optimal source codes for geometrically distributed integer alphabets, IEEE Trans. Inform. Theory March 1975 228–230.
[4] S.W. Golomb, Run length encodings IEEE Tran. Informat. Theory IT-12 (1966) 399–401.
[5] R. Hassan, A dichotomous search for a geometric random variable, Oper. Res. 32 (2) (1984) 423–439.
[6] D.A. Huffman, A method for the construction of minimum-redundancy codes, Proc. IRE 40 (1952) 1098–1101.
[7] P. Humblet, Optimal source coding for a class of integer alphabets, IEEE Trans. Inform. Theory IT-24 (1) (1978) 110–112.
[8] F.K. Hwang, On finding a single defective in binomial group testing, J. Amer. Statist. Assoc. 69 (345) (1974) 146–150.
[9] A. Kato, Huffman-like optimal prefix codes and search codes for infinite alphabets, Manuscript, January 20, 1997.
[10] A. Kato, Te Sun Han H. Nagaoka, Huffman coding with an infinite alphabet, IEEE Trans. Inform. Theory 42 (3) (1996) 977–984.
[11] T. Linder, V. Tarokh, K. Zeger, Existence of optimal prefix codes for infinite source alphabets, IEEE Trans. Inform. Theory 43 (6) (1997) 2026–2028.
[12] N. Merhav, G. Seroussi, M.J. Weinberger, Optimal prefix codes for two-sided geometric distributions (Abstract), Proc. Internat. Sympos. on Information Theory, 1997, p. 71.
[13] N. Merhav, G. Seroussi, M.J. Weinberger, Universal probability assignment in the class of two-sided geometric distributions (Abstract), Proc. Internat. Symp. on Information Theory, 1997, p. 70.
[14] R. Sedgewick, Algorithms, Addison-Wesley, Reading, MA, 1983.
[15] Y.C. Yao, F.K. Hwang, On optimal nested group testing algorithms, J. Statist. Plann. Inference 24 (1990) 167–175.